within the (or internal to) key switch **202** and this internal routing arrangement (not shown) may, in some cases, decrease potential ESD problems. Also, in an example embodiment, carbon or other electrically conductive traces may be provided on the PCB beneath the key switch **202**, or below dome post **220** to transfer signals from touchpad **204** to Mylar sheets **224**, for example.

[**0028**] Due to movement or travel of key **200** (or key switch **202**), which may be 0.1 mm or more, for example (e.g., 0.1 mm. 0.2 mm. 0.3 mm, 1 mm or 2 mm or more in some embodiments), flexible conductor **218**, may include folds, bends, coils, curves or other structure to allow for flexing or bending of flexible conductor **218** during travel of key switch **202**. Signals from keyboard controller **102** may be provided via wired or wireless communication to a host processor on a PC, laptop computer, cell phone, PDA, or other device, for example, for processing and/or display.

[**0029**] According to an example embodiment, additional control (e.g., hardware, software, and/or firmware which may be executed by keyboard controller **102** for example) may be provided to selectively enable and/or disable key input and/or touchpad (or touch) input. For example, a user may be typing and a finger may slide or move on touchpad **112** and this touch movement may be misinterpreted as a mouse or pointer movement. Therefore, according to one embodiment, a control system (e.g., software or firmware executed by keyboard controller **102**, a host processor, or other controller) may disable touchpads **112** and **114** during typing, and may enable the touchpads after a predetermined period of time since a last keystroke. Also, in one embodiment, keystrokes (or key input) may be ignored if they occur within a given time period since touchpad activity was detected. Such a control system is not required, but may be helpful in some cases in reducing inadvertent pointer movements from the touchpad caused by, for example, unintentional index finger motion on the touch sensors or touchpads (**112**, **114**). The operation of two additional example keyboard control systems are described in FIGS. **3** and **4**.

[**0030**] FIG. **3** is a flow chart **300** illustrating operation of a keyboard control system according to an example embodiment. First, it is determined if a key has been pressed (**302**). It is also determined if touch activity (e.g., at least slight pressure on a touchpad or touch sensor) is present (or has been detected) (**304**). If a key has been pressed, then key input may be enabled and touchpad (or touch) input may be disabled for a predetermined period of time after each key has been pressed (**306**). Otherwise, if a key has not been pressed, then the touch input may be enabled (**308**A). Also, in another example embodiment, if a key has not been pressed, then key input may also be disabled if touch activity is detected (**308**B).

[**0031**] FIG. **4** is a flow chart **400** illustrating operation of a keyboard control method according to another example embodiment. A current key state may be determined (**402**) and may be buffered for subsequent comparison. In an example embodiment, key state may be determined periodically or every t seconds, for example. Key states may include, for example, down (pressed) and up (not pressed) states or positions. If the key is down, then flow proceeds to **404**. The previous key state is determined (**404**). If the previous key state was down, then this indicates that the key is still down, and flow continues (**414**) back to **402**. If the previous key state was up (**404**) (and current key state is down), this indicates that the key was just pressed and flow waits at **406** for a predetermined period of time before disabling touch input (**408**). The touch input (e.g., touchpad and/or touch sensor) may be disabled for a predetermined period of time after the key was pressed (shown as wait, **412**).

[**0032**] Therefore, according to an example embodiment, when current key state is down, and previous key state was up, this may indicate that the keyboard is in the key mode (keystrokes), and may cause the keyboard to disable touch input for a predetermined period of time. However, the keyboard controller **102** (or other controller or processor), rather than disabling touch input, may merely temporarily buffer such touchpad and touch sensor input data (**410**) while keyboard controller **102** (or other controller or processor) may receive additional data (e.g., keystrokes, touchpad/touch sensor input or other information) to confirm that the keyboard should be in the key mode. Buffered data may, for example, either be used if appropriate, or may be discarded if not needed. Keyboard controller **102** may also consider by statistical methods or previous keyboard data, generally or for this specific user, to make decisions regarding automatic mode selection (e.g., determining when the key input or touch input should be enabled and disabled). Therefore, if keyboard controller **102** made an incorrect decision at **408** (e.g., it is determined that automatic mode selection selected an incorrect mode based on further data received by controller **102**), the touch data may be enabled and the buffered data (buffered at **410**) may be supplied to the host processor for processing or display.

[**0033**] If at **402** the current key state is up, then the previous key state is determined (**416**). If the previous key state was down, then this indicates that the key was just released, and the controller may wait (**418**) for a predetermined period of time before enabling touch input (**420**). These touch inputs may be enabled for a predetermined period of time (**422**), and then the process continues (**414**) back to **402**.

[**0034**] If at **402** the current key state is up, and at **416** the previous key state was up, then this indicates that the key is still up. It is determined if any touch activity is occurring (**424**). If no touch activity is occurring, then the flow continues (**414**) back to **402**. If significant touch activity is occurring, then the controller waits at **426**, then discards any received keystrokes (disables key input) for a predetermined period of time (wait **432**). In another embodiment, controller **102** may buffer (**430**) key (keystroke) input and confirm the keyboard mode that was selected. In this case, the controller **102** may receive further data or analyze statistical data, etc. to confirm that the keyboard mode selected (touch mode) was correct. If it was not correct (e.g., indicating that the keystroke data should not have been discarded), then the buffered keystroke data (buffered at **430**) may be provided to the host processor for processing or display. The method described in FIG. **4** may be implemented, for example, in keyboard controller **102**, a host processor, or some other processor or controller.

[**0035**] Implementations of the various techniques described herein may be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. Implementations may implemented